UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/631,930	07/31/2003	Robert M. English	112056-0112	9670

24267 7590 06/25/2007

CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210

EXAMINER	
ROMANO, JOHN J	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
06/25/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/631,930	ENGLISH ET AL.	
	Examiner	Art Unit	
	John J. Romano	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 April 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *(See attached)*.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's amendment and response received April 16th, 2007, responding to the January 16th, 2007, Office action provided in the rejections of claims 1-33, wherein claims 34-37 are added and claims 1-37 are pending in the application and which have been fully considered by the examiner.

Applicant arguing for the claims being patentable over *the prior art* (see pages 12-15 of the amendment and response) are not persuasive, as will be addressed under Prior Art's Arguments – Rejections section at item 2 and the claim rejections below. Thus, the rejection of the claims over prior art in the previous Office action is maintained in light of additional new grounds of rejection as necessitated by amendment and **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Prior Art's Arguments – Rejections

2. Applicant's arguments filed April 16th, 2007, in particular on pages 13-15, have been fully considered but they are not persuasive. For example,

(A) In regard to the argument that *Gillespie* does not teach Applicant's claimed invention because *Gillespie* teaches away from Applicant's processing of two concurrency groups at substantially the same time, (See response, page 14, 1st paragraph), the examiner respectfully disagrees. As referred to by Applicant (Page 14, 1st paragraph) and expressly disclosed by *Gillespie* (see abstract):

> "A scheduling kernel provides fair share scheduling of several virtual machines by a multi-processor scheduling module scheduling the virtual machines across the several processors of the multi-processor. A virtual machine scheduling module schedules threads of a virtual machine, and provides an independent scheduling policy for a virtual machine. Execution exclusion sets may be created and enforced by an execution exclusion set module to limit execution to a single thread at a time out of any particular execution exclusion set of threads. (See abstract, emphasis added).

Here, A particular virtual machine out of a plurality of virtual machine has an independent scheduling policy, wherein an exclusion set of threads may be enforced to limit execution to a single thread at a time. The plurality of virtual machines (execution vehicles) each having an independent scheduling policy, wherein the independent virtual machine each may create and enforce the exclusion set of threads (concurrency groups). Therefore, although *Gillespie* does not expressly discloses processing two execution sets at one time he certainly suggests it by teaching multiple virtual machines

(execution vehicles) executing in parallel in a multi-processor system having independent scheduling policies for their respective threads.

In regard to Applicant's argument that there is a clear teaching away in *Gillespie* because the execution exclusion sets may be limited to a single particular thread out of all the execution exclusion sets (See response, page 14, second paragraph) the examiner respectfully disagrees. As referred to by the Applicant and recited above, *Gillespie* teaches a single thread out of any particular set of threads rather than "a single particular thread out of all the execution exclusion sets" as argued by Applicant. Further evidence is taught by *Gillespie* (See column 4, lines 59-65), reproduced below for convenience:

> "In one currently preferred embodiment 10 of an apparatus and method in accordance with the invention, a group of threads may be assembled into a virtual machine. The virtual machine may be scheduled by a multi-processor scheduling module 22. The multi-processor scheduling module 22 may itself be instantiated in several instances, each associated with one processor 12 (12a, 12b, or 12c) of the several processors 12a, 12b, 12c available in a multi-processor 10 or a multi-processing environment 10".

Also of interest is *Gillespie's* teaching that each one of the virtual machines by a multiprocessor scheduling module 22 may have an independent policy (Column 3, lines 1-15). In other words, *Gillespie's* teaches an exclusion set associated with a particular virtual machine, wherein a particular virtual machine may be associated with a single processor of a plurality of processors allowing the other processors associated with the other virtual machines independent scheduling at substantially the same time.

Accordingly, *Gillespies's* exclusion set is the equivalent to Applicant's concurrency group with the equivalent function, as taught by *Gillespie* "the execution set module provides for consistent processing of non-parallel processable threads without funneling" in a multi-processor system (See *Gillespie*, Column 14, lines 16-25). Therefore, it is at least obvious that Gillespie teaches two virtual machines (execution sets) executing an exclusion set (concurrency group), wherein the threads of each exclusion set are restricted from running in parallel.

Thus, the rejection of the claims is maintained in light of Applicant's instant argument with respect to the independent claims. Accordingly, the dependent claims are also maintained for the reasons above and below in the claim rejections.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims **1-37** are rejected under 35 U.S.C. 103(a) as being unpatentable over Gillespie, US 6,269,391 (hereinafter **Gillespie**).

In regard to claim **1**, **Gillespie** discloses

- *"A method for executing uniprocessor (UP) coded workloads in a multiprocessor (MP) computer system without having to rewrite the*

UP-coded workloads' code..." (E.g., see Figure 5 & Column 14, lines 16-25), wherein non-parallel processable threads can be executed via the exclusion set (concurrency group) method in a multiprocessing system.

- "*...organizing the UP-coded workloads into one or more concurrency groups, wherein UP-coded workloads in the same concurrency group are not permitted to execute concurrently with one another in the MP computer system...*" (E.g., see Figure 4 & Column 6, lines 48-56), wherein an execution exclusion set membership 142 excludes parallel or concurrent execution.
- "*...scheduling first and second execution vehicles that respectively execute on different processors in the MP computer system at substantially the same time...*" (E.g., see Figure 1 & Column 4, lines 59-67), wherein the multi-processor scheduling module 22 may itself be instantiated in several instances (execution engine), each associated with one processor 12 (12a, 12b, 12c) available in a multi-processor environment.
- "*...acquiring a first concurrency group by the first execution vehicle and a second concurrency group by the second execution vehicle; and executing UP-coded workloads in the first concurrency group through the first execution vehicle at substantially the same time as UP-coded workloads in the second concurrency group are executed through the*

second execution vehicle." (E.g., see Figure 5 & Column 13, lines 44-47), wherein each processor may be bound to an exclusion set.

But **Gillespie** does not expressly disclose "a second concurrency group by the second execution vehicle". However, it would have been that a second concurrency group or instance of another virtual machine to perform in parallel on the multiprocessing system. This is evident by Gillespie's express teaching of multiple instances of Virtual machine objects (e.g., a, b, c..., see Figure 2 (28) & Column 8, lines 5-15), wherein a specific thread control object can be bound to one of the specific processors 12a, b, c, wherein the other processors could obviously be bound to a different or second thread control block. Thus, it would have been obvious to achieve the benefits known in the art of multiprocessing to have a second execution vehicle acquire a second thread control block which may be a member of a exclusion set.

In regard to claim **2**, the rejections of base claim **1** are incorporated. Furthermore, **Gillespie** discloses:

- "...*the UP-coded workloads are UP-coded threads, and the first and second execution vehicles are first and second processes.*" (E.g., see Figure 5 & Column 13, lines 57-61), wherein each instance of the kernel scheduling process controls the execution of the respective threads.

In regard to claim **3**, the rejections of base claim **1** are incorporated. But, **Gillespie** does not expressly discloses "...*the UP-coded workloads are messages, and the first and second execution vehicles are first and second threads.*". However, it

would have been obvious to one of ordinary skill in the art, at the time the invention was made, to employ the UP-code workloads as messages, and the execution vehicles as threads. The motivation would have been to employ messages as is old and well known in the art of parallel programming and network environments, wherein a message is communication data received and processed by an execution environment (computer) and thus would have been obvious to group messages into a particular execution environment.

In regard to claim **4**, the rejections of base claim **1** are incorporated. Furthermore, **Gillespie** discloses:

- "*...dequeueing from a concurrency-group run queue a first concurrency-group data structure associated with the first concurrency group; and dequeueing from the concurrency-group run queue a second concurrency-group data structure associated with the second concurrency group.*" (E.g., see Figure 3 & Column 8, lines 16-42), wherein CPU-specific scheduler 64 comprises a scheduling queue 98 linked to a specific data segment.

In regard to claim **5**, the rejections of base claim **4** are incorporated. Furthermore, **Gillespie** discloses:

- "*...setting a first CG flag in the first concurrency-group data structure to a value indicating that the first concurrency group is in a running state....*" (E.g., see Figure 5 & Column 13, lines 30-34), wherein the occupied lock 154 indicates if a thread is executing.

In regard to claim **6**, the rejections of base claim **4** are incorporated. Furthermore, **Gillespie** discloses:

- "*...appending UP-coded workloads enqueued on a first current queue in the first concurrency-group data structure onto a first active queue in the first concurrency-group data structure...*" (E.g., see Figure 4 & Column 10, lines 14-24), wherein the CPU specific data comprises an active pointer link 112 and a current lock status (flag) wherein the ready queue 128 is also disclosed.

In regard to claim **7**, the rejections of base claim **6** are incorporated. Furthermore, **Gillespie** discloses:

- "*...dequeueing UP-coded workloads in the first ... concurrency groups from the first ... active queues...; and executing the dequeued UP-coded workloads to completion.*" (E.g., see Figure 5 & Column 13, lines 6-15), wherein a waiting queue 160 is dislcosed.

In regard to claim **8**, the rejections of base claim **5** are incorporated. Furthermore, **Gillespie** discloses:

- *"...A) if at least one UP-coded workload in the first concurrency group is executable (i) setting the value of the first CG flag to a value indicating that the first concurrency group is in a queued state...*" (E.g., see Figure 6 & Column 13, lines 48-56), wherein the particular thread must obtain approval in order to execute, such as by setting a flag, key, lock or the like..

- "...*(ii) re-enqueueing the first concurrency-group data structure onto the concurrency-group run queue...*" (E.g., see Figure 2 & Column 3, line 53-Column 4, line 4), wherein the context switch will reschedule the thread for later (re-enqueing).
- "...*B) if there are not any UP-coded workloads in the first concurrency group that are executable, setting the first CG flag to a value indicating that the concurrency group is in a suspended state.*" (E.g., see Figure 2 & Column 4, lines 13-26), wherein a suspended state is taught.
- "...*C) dequeueing from the concurrency-group run queue a third concurrency-group data structure associated with a third concurrency group; and D) setting a third CG flag in the third concurrency-group data structure to a value indicating that the third concurrency group is in a running state.*" (E.g., see Figure 4, box 130 + 132 & Column 8, lines 25-28), wherein the context of the thread control object is stored 130 and the link to the next control.

In regard to claim **9**, the rejections of base claim **1** are incorporated. Furthermore, **Gillespie** discloses:

- "...*the UP-coded workloads is organized into the one or more concurrency groups at run-time.*" (E.g., see Figure 5 & Column 13, lines 35-43), wherein the membership in an execution exclusion set may be transitory, subject to certain execution options represented in the data.

In regard to claim **10**, the rejections of base claim **1** are incorporated. But **Gillespie** does not expressly "network cache". However, it would have been obvious to one of ordinary skill in the artl, at the time of the invention, to implement the MP system as a network cache as a network cache is old and well known in the multiprocessing art. Furthermore, **Gillespie** discloses multiple caches (Figure 1, see box 14a, 14b, 14c).

In regard to claim **11**, **Gillespie** discloses:

- "*A multiprocessor (MP) computer system configured to execute uniprocessor (UP) coded threads without having to rewrite the UP-coded threads' code, the MP computer system comprising: a plurality of processors; a memory having a plurality of storage locations addressable by the plurality of processors for storing data and program code...*" (E.g., see Figure 1).
- "*...the memory being configured to store a separate concurrency-group data structure for each of a plurality of concurrency groups, each concurrency-group data structure ...*" (E.g., see Figure 1 & Column 14, lines 16-25), wherein non-parallel processable threads can be executed via the exclusion set (concurrency group) method in a multiprocessing system.
- "*...an active-queue pointer storing a location in the memory of an active queue of UP-coded thread messages associated with UP-coded threads in an executable state...*" (E.g., see Figure 4 & Column 10, lines 14-24), wherein the CPU specific data comprises a active pointer

link 112 and a current lock status (flag) wherein the ready queue 128 is also disclosed.

- "*...and a current-queue pointer storing a location in the memory of a current queue of UP-coded thread messages associated with UP-coded threads waiting to be transferred to the active queue.*" (E.g., see Figure 5 & Column 13, lines 6-15), wherein a waiting queue 160 is dislcosed.

In regard to claim **12**, the rejections of base claim **11** are incorporated. Furthermore, **Gillespie** discloses:

- "*...a CG flag that stores a value indicating an operational state of a concurrency group associated with the concurrency-group data structure.*" (E.g., see Figure 5 & Column 13, lines 30-34), wherein the occupied lock 154 indicates if a thread is executing.

In regard to claim **13**, the rejections of base claim **11** are incorporated. Furthermore, **Gillespie** discloses:

- "*...each UP-coded thread message stored in the active queue and current queue stores a location in the memory of a top of a call stack associated with a specific UP-coded thread.*" (E.g., see Figure 4 (158 + 160) & Column 13, lines 6-15), wherein a recursive call stack , including context and link (132) for the next ready threads 160.

In regard to claim **14**, the rejections of base claim **13** are incorporated. Furthermore, **Gillespie** discloses:

- "*...the call stack is accessible through a thread control block (TCB) associated with the specific UP-coded thread, the TCB including a CG pointer for storing a memory location of a concurrency-group data structure.*" (E.g., see Figure 4 (30) & Column 13, lines 6-15), wherein a thread control object (TCO) 30 with a recursive call stack , including context and link (132) for the next ready thread 160.

In regard to claim **15**, the rejections of base claim **11** are incorporated. Furthermore, **Gillespie** discloses:

- "*...each concurrency-group data structure further comprises meta-data information associated with a concurrency group.*" (E.g., see Figure 5), wherein recursive depth counter 158 and thread ID 158 is metadata.

In regard to claim **16**, see claim **10**.

In regard to claims **17-24**, this is an apparatus version of the claimed method discussed above, in claims **1-8**, respectively, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see **Gillespie**, storage device (Figure 1 & Column 3, lines 56-60), wherein instructions to implement the process may be stored.

In regard to claims **25-27**, this is a computer readable media version of the claimed method discussed above, in claims **1-3**, respectively, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see

Gillespie, storage device (Figure 1 & Column 3, lines 56-60), wherein instructions to implement the process may be stored.

In regard to claims **28-33**, this is another method version of the claimed method discussed above, in claims **1** and **4-8,** respectively, wherein all claimed limitations have also been addressed and/or cited as set forth above.

In regard to claims **34-37**, this is another method version of the claimed method discussed above, in claims **1, 2, 3** and **1,** respectively, wherein all claimed limitations have also been addressed and/or cited as set forth above

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John J. Romano whose telephone number is (571) 272-3872. The examiner can normally be reached on 8-5:30, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JJR

TUAN DAM
SUPERVISORY PATENT EXAMINER